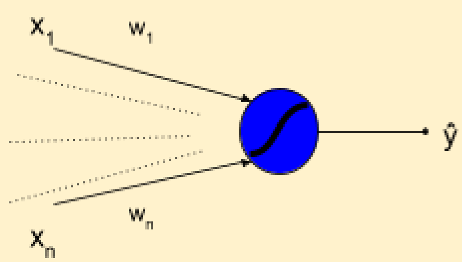
PadhAI: Representation Power of Functions

One Fourth Labs

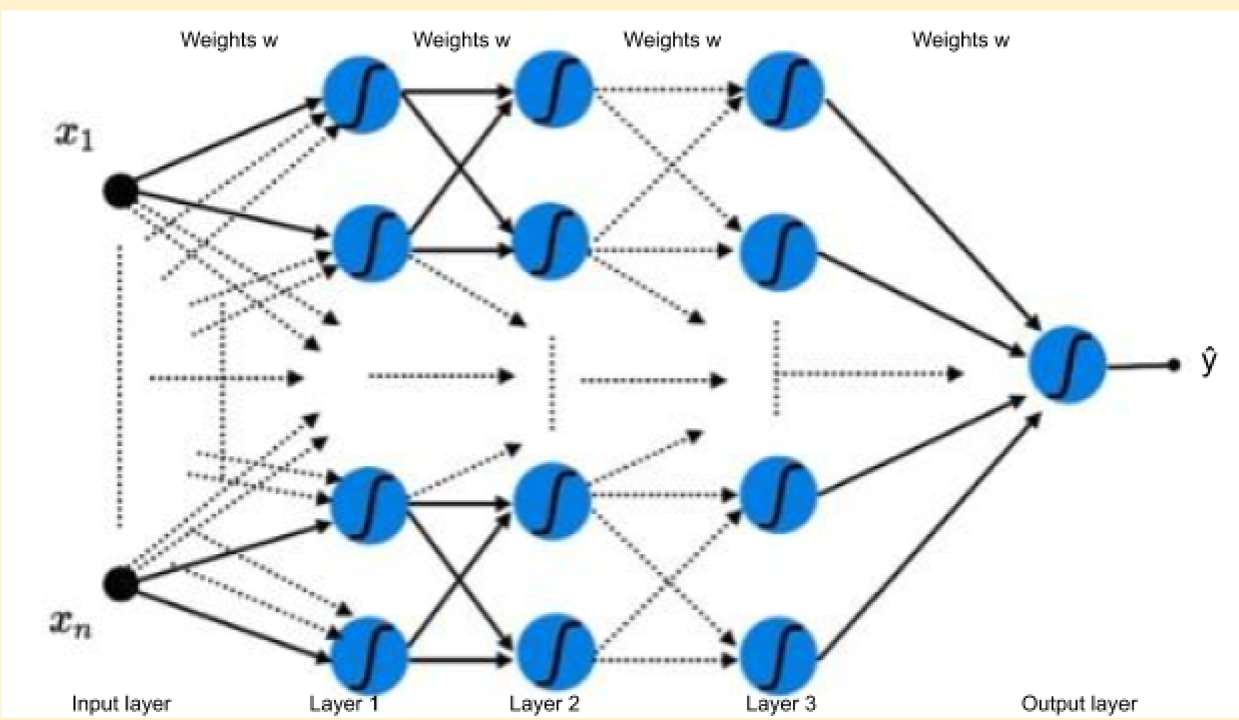
Building complex functions (A simple recipe)

How do we even come up with such functions?

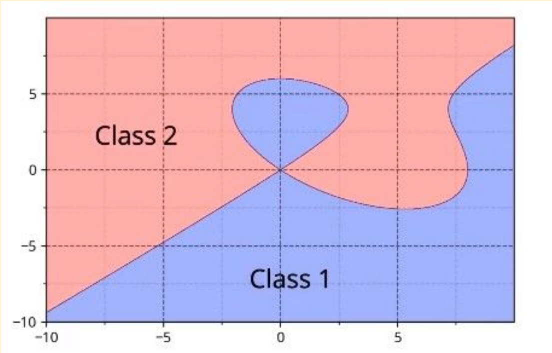
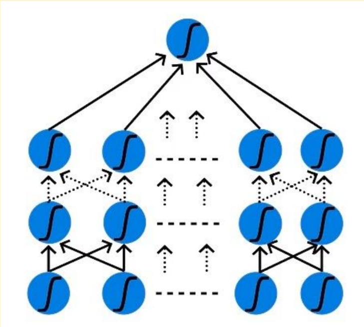
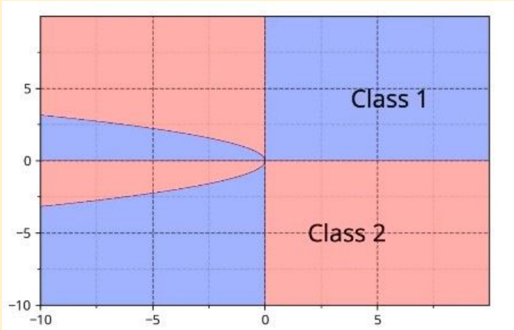
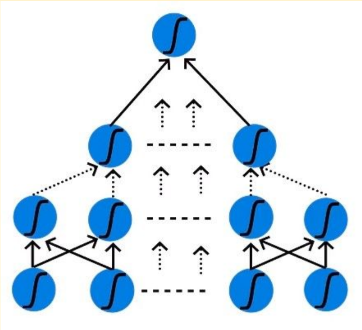
1. $\hat{y} = (\sin wx + \cos wx^3 + e^x + x^{10}) * \frac{1}{log\, x}$ is an example of a function that could create a complex decision boundary
2. Clearly, we can see that it's hard to come up with such functions, thus we need a simpler approach
3. Consider the following analogy, to build a house/building, we don't simply conjure the building out of thin air, instead we consider the most basic unit of the building: the brick.
4. The bricks are combined one after the other, in different ways, that ultimately amount to a very complicated structure.
5. In our context, the building would be the complex function and the brick would be a single sigmoid neuron
6. So, here's the brick



7. And here's the building



8. The building that we have constructed with the sigmoid neurons is nothing but a **deep neural network**.
9. Consider a complex function **y’** (read y-prime)
10. The output function of the deep neural network $\hat{y} = f(x_1, x_2, ...x_n)$
11. Regardless of what **y’** we consider, we will be able to approximate it with ŷ by using different configurations of layers and sigmoid neurons.
12. To state this more formally: A deep neural network with a certain number of hidden layers would be able to approximate any function between the input and output
13. This is called the Universal Approximation Theorem **(ŷ≃y’)**
14. Consider the examples from the previous section

Complex function	Deep Neural Network Representation
	
	

15. With regards to figuring out the DNN configuration for each function, we have to try out different combinations to see what fits best
16. For eg:
    a. Select between 1 - 7 hidden layers
    b. Each hidden layer can have 50, 100 or 150 neurons
    c. Construct several neural networks and select the combination that yields the minimum loss
    d. Thanks to the democratization of models, we have a fairly good idea of the combinations to select based on the task at hand, ie we don’t need to try all possible configurations.